

27장 배열

정의

배열이란 여러개의 값을 순차적으로 나열한 자료구조

구조

1. 요소

- 배열이 가지고 있는 값
- js의 모든값은 배열의 요소가 될수 있습니다.

2. 인덱스

- 자신의 위치를 나타내는 0 이상의 정수
- 배열의 요소를 접근할때 사용합니다.

3. length

- 요소의 개수 즉 배열의 길이를 나타냄

특징

배열은 객체지만 일반 객체와는 구별되는 독특한 특징이 있다.

구분	객체	배열
구조	프로퍼티 키 와 프로퍼티 값	인덱스와 요소
값의 참조	프로퍼티 키	인덱스
값의 순서	X	O
length 프로퍼티	X	O

- 가장 명확한 차이는 값의 순서와 length 프로퍼티 이다.
- 인덱스로 표현되는 값의 순서와 length 프로퍼티가 존재함으로써

1. 반복문을 통한 접근
2. 순차적인 접근
3. 역순 접근
4. 특정 위치로부터의 순차적인 접근

에 유리한 특징을 가지고 있다.

자바스크립트 배열은 배열이 아니다.

기존의 자료구조에서 표현하는 배열은

1. 동일한 크기의 메모리 공간이
2. 동일한 데이터타입을 가지는 요소로
3. 빈틈없이
4. 연속적으로
5. 나열된

자료구조를 말한다.

이러한 배열을 **밀집배열** 이라 한다.

이러한 의미의 배열은 각 요소가 동일한 데이터 크기를 가지며 빈틈없이 연속적으로 이어져 있으므로 인덱스를 통해 단 한번의 연산으로 임의의 요소에 접근 할 수 있다.

자바스크립트 에서의 배열은 일반적인 의미의 배열과 다르다.

- 배열의 요소를 위한 각각의 메모리 공간이 동일한 크기를 갖지 않아도 되며
- 연속적이지 않을수도 있다.

이러한 배열을 **희소배열** 이라고 한다

해당 구조를 채택함으로써, 자바스크립트의 배열은 인덱스로 접근하는 경우의 성능보다는 특정 요소를 탐색하거나 배열 요소를 삽입 또는 삭제하는 경우의 성능을 우선시한 것이라고 할 수 있습니다.

length 와 희소 배열

`length` 속성은 배열의 요소가 변경될 때 자동으로 갱신됩니다. 그러나 일부 요소가 비어 있을 때 `length` 와 실제 배열 요소의 개수가 일치하지 않을 수 있습니다.

이는 배열의 기본적인 개념인 연속적인 값의 집합과 맞지 않으며, 성능에도 부정적인 영향을 줄 수 있습니다.

따라서 배열에는 같은 타입의 요소를 연속적으로 위치시키는 것이 최선입니다.

배열 생성

배열의 다양한 생성 방식

1. 배열 리터럴

```
const arr = [1, 2, 3];
console.log(arr.length); // 3
```

2. Array 생성자 함수

```
const arr = new Array(10);
console.log(arr); // [empty × 10]
console.log(arr.length); // 10
```

3. Array.of (ES6)

- 전달된 인수를 요소로 갖는 배열을 생성

```
Array.of(1); // -> [1]
Array.of(1, 2, 3); // -> [1, 2, 3]
Array.of('string'); // -> ['string']
```

4. Array.from (ES6)

- 유사 배열 객체 또는 이터러블 객체를 인수로 전달받아 배열로 변환하여 반환

```
// 유사 배열 객체를 변환하여 배열을 생성한다.
Array.from({ length: 2, 0: 'a', 1: 'b' }); // → ['a', 'b']
```

```
// 이터러블을 변환하여 배열을 생성한다. 문자열은 이터러블이다.  
Array.from('Hello'); // → ['H', 'e', 'l', 'l', 'o']
```

배열 요소의 참조

배열의 요소를 참조할 때에는 대괄호 표기법([])을 사용한다.

```
const arr = [1, 2];  
  
// 인덱스가 0인 요소를 참조  
console.log(arr[0]); // 1
```

- 인덱스를 나타내는 문자열을 프로퍼티 키로 갖는 객체다.
- 따라서 존재하지 않는 프로퍼티 키로 객체의 프로퍼티에 접근했을때 `undefined`를 반환하는 것 처럼
- 배열도 존재하지 않는 요소를 참조하면 `undefined`를 반환한다.

배열 요소의 추가와 갱신

배열에도 요소를 동적으로 추가할 수 있다.

이때 length 프로퍼티 값은 자동으로 갱신된다.

```
const arr = [0];  
  
// 배열 요소의 추가  
arr[1] = 1;  
  
console.log(arr); // [0, 1]
```

배열 요소의 삭제

배열의 특정 요소를 삭제하기 위해서는 delete 연산자를 사용할 수 있다.

```
const arr = [1, 2, 3];

// 배열 요소의 삭제
delete arr[1];
console.log(arr); // [1, empty, 3]

// length 프로퍼티에 영향을 주지 않는다. 즉, 희소 배열이 된다.
console.log(arr.length); // 3
```

하지만 삭제 후에도 length 프로퍼티에 영향을 주지 않는다
따라서 희소배열을 만들지 않으면서 특정요소를 완전히 삭제하려면

`Array.prototype.splice` 메서드를 사용해야한다.